European Journal of Operational Research 51 (1991) 400–404
North-Holland

Theory and Methodology

Data dependent worst case bound improving techniques in zero–one programming

Osman Oğuz
Department of Industrial Engineering, Bilkent University, Ankara, Turkey

Abstract: A simple perturbation of data is suggested for use in conjunction with approximation algorithms for the purpose of improving the available bounds (upper and lower), and the worst case bounds. The technique does not require the approximation algorithm (heuristic) to provide a worst case bound to be applicable.

Keywords: Zero–one programming, knapsack, heuristics

1. Introduction

Fisher [2] points out to the fact that including data dependent parameters in the worst case analysis of approximation algorithms may lead to better bounds of performance. He demonstrates this on a simple knapsack problem and suggests a new research avenue on this line. We pick up on his suggestion and develop a technique of incorporating data dependent information in the worst case performance analysis of some heuristic algorithms designed to solve the multidimensional knapsack problem.

First we summarize an approximation algorithm for the multidimensional knapsack problem which is an extension of Sahni's algorithm [8] for the single dimensional case. Then we show how the information provided by the approximate solution may be used in narrowing the search space by using a stability concept developed previously by Oguz and Magazine [6]. The implications of this concept is discussed shortly in relevance with the approximation algorithm mentioned above. We introduce the data perturbation technique after this and show how it can be employed to improve the bounds obtained by the application of the approximation algorithm alone. Finally, a brief discussion on potential use of the perturbation technique together with other heuristics or exact algorithms is given.

Received February 1989; revised August 1989

2. The approximation algorithm

The problem under consideration is:

$$\begin{aligned} \text{Max.} \quad & \sum_j^n c_j x_j \\ \text{s.t.} \quad & \sum_j^n a_{ij} x_j \leqslant b_i, \quad \forall i = 1, \dots, m, \\ & x_j = \{0, 1\}, \quad \forall j = 1, \dots, n. \end{aligned} \tag{1}$$

We assume that all parameters have positive integer values. The ε-approximation algorithm is a combination of partial enumeration and LP (linear programming) relaxation. Since it is now possible to solve the LP relaxations in polynomial

0377-2217/91/$03.50

time [4], the algorithm which will be presented is also polynomial.

Let $z(L)$ denote the objective function value of the best solution found by the algorithm. The algorithm provides this solution by totally enumerating all combinations of variables with size less than or equal to L. A solution is obtained by setting the variables in the chosen combination equal to 1 and then solving a LP relaxation over the remaining variables and resources (i.e., the right hand side vector b is also adjusted). The indices of variables which are set equal to one in the LP relaxation constitute the set S. Other variables, including the ones with fractional values, are equal to zero. The value of such a solution is obviously equal to

$$\sum_{j\in I} c_j + \sum_{j\in S} c_j, \tag{2}$$

where I is the index set of variables in the chosen combination. Thus,

$$z(L) = \sum_{j\in I} c_j + \sum_{j\in S} c_j, \tag{3}$$

for a specific I and S, and no other choice of I with $|I| \leqslant L$ can provide a better solution with this algorithm. There can be at most m variables with fractional values in the LP relaxation. The rest will be either zero or one. We can use this fact to slightly extend Sahni's [8] result for the single dimensional case to show, as was done in [6], that

$$z(L) \geqslant (L/(L+m))z^*, \tag{4}$$

where z^* is the value of an optimal solution to problem (1). The proof of this is based on the fact that in enumerating all combinations of size L, we will encounter those variables with the largest c_j's in an optimal solution. That is, the set I will consist of these variables only. At this moment we will have

$$z^* \leqslant \sum_{j\in I} c_j + \sum_{j\in S} c_j + mc_{\max}. \tag{5}$$

We can assume $c_{\max}$ is equal to the smallest c_j such that $j \in I$. This means $c_{\max} \leqslant (1/L)z(L)$. Thus, we can write

$$\begin{aligned} z^* &\leqslant z(L) + m(1/L)z(L) \\ &= ((L+m)/L)z(L), \end{aligned} \tag{6}$$

which in turn gives $z(L) \geqslant (L/(L+m))z^*$. Of course, L has to be larger than m to have a worst case bound of at least $\frac{1}{2}$, which may be prohibitively expensive in terms of computational load for large m.

3. The stable sets

We have developed a concept of stability in [7]. The idea is a generalization of the well-known variable fixing technique originally proposed by Ingargiola and Korsh [3].

Referring back to problem (1) again, we assume that an optimal solution of the LP relaxation of this problem is found using the upper bounded variables version of the simplex algorithm. Again, let S denote the index set of variables which are equal to one in this solution (in the discussions to follow, the set I is assumed to be empty since we are not concerned with enumeration). S may be empty, which simply means $z^{\ell} = 0$. $z^{\ell} = \sum_{j\in S} c_j$ is an obvious lower bound, and an upper bound z^{u} is the optimal objective functions value of the LP relaxation. We use $\bar{c}_j$, $j = 1, \ldots, n$, to denote the reduced costs of the 0–1 variables in the simplex tableau, so that $\bar{c}_j \geqslant 0$, $\forall j \in S$ and $\bar{c}_j \leqslant 0$, $\forall j \notin S$.

We adopt the following definition and propositions from [7].

Definition. A set J is *stable* if $\sum_{j\in J} |\bar{c}_j| \geqslant z^{\mathrm{u}} - z^{\ell}$ and the same is not true for any proper subset of J. The smallest integer K, such that J or some subset of J is stable for all $J \subseteq N = \{1, \ldots, n\}$ and $|J| \leqslant K$ is called the *stability number* of problem (1).

Proposition 1. *Let $J \subset N$ with $|J| \geqslant K$, $T = N \setminus S$, and $X = \{x_1, \ldots, x_n\}$ be an optimal solution. Then*

$$\sum_{J\cap S} (1 - x_j) + \sum_{J\cap T} x_j \leqslant K - 1.$$

Proof. If this does not hold, $CX \leqslant z^{\ell}$ holds, which is a contradiction.

The stability number K can easily be determined by solving the following simple single dimensional knapsack problem:

$$\begin{aligned} K = \text{Max.} \quad & \sum_{i\in N} y_i + 1 \\ \text{s.t.} \quad & \sum_{i\in N} |\bar{c}_i| y_i \leqslant z^{\mathrm{u}} - z^{\ell}, \\ & y_i \in \{1, 0\}, \ \forall i = 1, \ldots, n. \end{aligned}$$

The LP relaxation of this problem will have at most one fractional valued variable. Setting that variable to zero gives the optimal integer solution, and thus K is determined.

Proposition 2. *$z^{\ell} \geqslant Rz^*$ where $R = (\lfloor |S|/K \rfloor)/(\lfloor |S|/K \rfloor + 1))$.*

Proof. $z^1 \geqslant \lfloor |S|/K \rfloor (z^{u} - z^{\ell})$ because there are at least $|S|/K$ disjoint stable subsets of S, and sum of $\bar{c}_j$ for each subset is at least as great as $z^{u} - z^{\ell}$, and $z^{\ell} = \sum_{j \in S} c_j \geqslant \sum_{j \in S} \bar{c}_j$. Also $z^{u} \geqslant z^{\ell}$ and $|S|/K \geqslant \lfloor |S|/K \rfloor$, which means the above is true.

We would like to note here that K may be too large or even undefined (i.e., $K = n + 1$ means there is no stable set, so we may assume that K is undefined for this case) for some problems. The subset sum problems, a special case of the knapsack problem where all $c_j = a_j$, are good examples of these. This is not in contradiction with the concepts developed in this study since 'data dependence' is the underlying motif. Let us now consider a slight revision of the ε-approximation algorithm described in Section 2. Assume that we have solved the LP relaxation, so that S and K are known. Choose $L \leqslant K - 2$. The algorithm:

Step 1. Set $i = 1$.

Step 2. Set a previously untested combination of i variables in S equal to zero. The remaining variables in S are equal to 1.

Step 3. Choose a previously untested combination of size $1, \ldots, \min(L, K - 2 - i)$ of variables not in S, and set then equal to one if feasible, and solve the LP relaxation over the remaining variables. If all possible combinations are exhausted, save the best solution found so far and go to Step 4.

Step 4. Set $i = i + 1$. If $i \leqslant L$, return to Step 2. Otherwise stop.

If we set $L = K - 1$, then the algorithm above finds an optimal solution, because the search process implicitely enumerates all possibilities in that case. That's why L is set less than or equal to $K - 2$. In the execution of the algorithm we will encounter a situation where $|S| - (K - 2) + L$ variables with the greatest c_j values in an optimal solution will be set equal to one. This means a worst case bound of

$$z(L) \geqslant ((|S| - (K-2) + L) / (|S| - (K-2) + L + m))z^*,$$

in terms of the ε-algorithm described in Section 2. But it is also true that

$$((|S| - (K-2) + L) / (|S| - (K-2) + L + m)) \geqslant (R + L)/(L + m)$$

for any $L \leqslant K - 2$. This can easily be verified by substituting $((|S|/K)/(|S|/K) + 1))$ instead of R in the above inequality. Thus, $z \geqslant ((R + L)/(L + m))z^*$ holds true. The significance of this fact will become much clearer when we relate R to data perturbation technique in the next section.

4. Data perturbation

Let us now consider perturbing the cost coefficients of problem (1) be setting

$$c_j' = c_j + p_j, \quad \forall j = 1, \ldots, n, \tag{7}$$

where

$$\sum_{j=1}^{n} p_j \leqslant \alpha, \quad p_j \geqslant 0, \quad j = 1, \ldots, n. \tag{8}$$

Then for any set $S \subset N$, we have:

$$\sum_{j \in S} (c_j' - c_j) \leqslant \alpha. \tag{9}$$

Suppose we would like to have $\alpha \leqslant \varepsilon z^*$, where z^* represents the value of an optimal solution. Choosing $\alpha = \varepsilon z^1$ satisfies this condition. Consider problem (1) with a perturbed objective function. If we can find the optimal solution of this problem, then we will obviously have

$$z^{*\prime} - z^{a} \leqslant \alpha \leqslant \varepsilon z^*, \tag{10}$$

or

$$z^{*\prime} - z^{a} \leqslant \varepsilon z^*,$$
$$z^* - z^{a} \leqslant \varepsilon z^*,$$
$$z^{a} \geqslant (1 - \varepsilon)z^*.$$

Where z^* is the optimal value of the perturbed objective function, z^{a} is the value of the original objective function corresponding to the optimal solution of the problem with the perturbed objective function, and z^* is the optimal objective function value of the original problem.

Let us now look at the implications of this on the approximation algorithm explained earlier. Supposing that the perturbed problem is solved using the approximation algorithm, the worst case bound in terms of original data obviously will be

$$z(L) \geqslant ((R+L)/(L+m))z^*(1-\varepsilon). \tag{11}$$

Here, at this point, it becomes natural to consider a trade off between the values of R and ε. Recall that value of R depends on K, the stability number. We can play with the value of K by our choice of ε. That is, we can even have $K=1$ if we choose ε large enough. Greater values of ε causes K to have smaller values, thus leading to larger R values. One good use of this fact is to strike a balance between R and ε to obtain the best possible bound. This can be especially practical for small values of m. For example, a specific case with $R=0.6$, $m=1$, and $L=2$ will have $z(L) \geqslant 0.87z^*$ approximately. If we can raise R to 0.99 by setting $\varepsilon = 0.05$, then the overall bound will be approximately $z(L) \geqslant 0.95z^*$.

Another possibility is to make a relatively difficult problem approximately solvable by decreasing K. An implication of Proposition 1 is that no more than $K-1$ variables can have different values in an optimal solution and the solution given by $x_i=1$, $\forall i \in S$, $x_i=0$ otherwise. This means that the optimal solution of the perturbed problem can be found by testing at most $(K-1)n^{(K-1)}$ solutions by any enumerative scheme. To see what we mean more clearly, consider the following simple 0–1 knapsack problem:

$$\begin{aligned} \text{Max.}\quad & z = 12x_1 + 10x_2 + 27x_3 + 16x_4 \\ & \quad + 11x_5 + 6x_6 \\ \text{s.t.}\quad & 2x_1 + 2x_2 + 6x_3 + 4x_4 + 4x_5 + 3x_6 \leqslant 12, \\ & x_j = \{0, 1\}, \quad j = 1, \dots, 6. \end{aligned} \tag{12}$$

The relaxed LP solution of this problem is:

$$x_1 = x_2 = x_3 = 1, \quad x_4 = 0.5, \quad x_5 = x_6 = 0,$$
$$\text{with} \quad z^{\text{u}} = 57.$$

The relative cost coefficients are

$$\bar{c}_1 = 4, \quad \bar{c}_2 = 2, \quad \bar{c}_3 = 3, \quad \bar{c}_4 = 0, \quad \bar{c}_5 = -5,$$
$$\bar{c}_6 = -6.$$

We also have

$$S = \{x_1, x_2, x_3\} \quad \text{with} \quad z' = 49.$$

According to our earlier definitions, the stability number $K=4$ for this problem. Let us change the objective function so that

$$c_1' = 12 + \delta, \quad c_2' = 12 + \delta, \quad c_3' = 28 + \delta,$$
$$c_4' = 16, \quad c_5' = 11, \quad c_6' = 6.$$

Assume that δ is negligibly small. The stability number K of the new problem is equal to 3. Thus, at most two variables can be complemented. That is, to find the optimal solution, at most two variables can be assigned valued different from their present values. One of these two has be the fourth variable because any pair excluding this variable is stable. So the optimal solution can easily be determined as:

$$x_1 = x_3 = x_4 = 1 \quad \text{with} \quad z = 56.$$

The objective function value of the original problem corresponding to this solution is 55 and the new upper bound on it is obviously 56. This is significant because it demonstrates that the perturbation technique can narrow the gap between upper and lower bounds by adjustments at both ends, that is, by lowering the upper bounds and raising the lower bounds at the same time.

As we have pointed out earlier, the technique can be used with any enumerative (search) algorithm whether approximate or exact. It can be used with Balas and Martin's [1] pivot and complement heuristic for instance, to reduce the size of the search space or to fix larger numbers of variables. Inaccuracy of data introduced by the used perturbations, will be much more than compensated for by the better quality of solutions obtained.

5. Conclusions

We have shown that if problem data is included in the analysis, it becomes possible to improve the worst case performance bounds of approximation algorithms. This has been achieved by using a concept of stability reflecting the relative difficulty of the problem under consideration together with data perturbations. Another important result is the possibility of obtaining upper bounds letter than those given by the LP relaxation, which is commonly regarded to be the best possible [5].

Further research to find a general method of making perturbations in some 'optimal' sense is needed. The extension of the stability concept and perturbation technique for other combinatorial problems is worth considering. Also, carrying out computational experimentation using the technique described in this study may give interesting results.

References

[1] Balas, E. and Martin, C., "Pivot and complement — A heuristic for 0–1 programming", *Management Science* 26 (1980) 86–96.

[2] Fisher, M., "Worst-case analysis of heuristic algorithms", *Management Science* 26 (1980) 1–17.

[3] Ingargiola, G. and Korsh, J., "Reduction algorithm for zero–one single knapsack problems", *Management Science* 20/4 (1974).

[4] Karmarkar, N., "A new polynomial-time algorithm for linear programming", *Combinatorica* 4 (1984) 373–395.

[5] Nemhauser, G. and Ullman, Z., "A note on the generalized Lagrange multiplier solution to integer programming problem", *Operations Research* 16/2 (1968).

[6] Oğuz, O. and Magazine, M., "A ε-approximate algorithm for the multidimensional 0–1 knapsack problem", Working paper, Department of Management Science University of Waterloo, 1981.

[7] Oğuz, O. and Magazine, M., "A stability concept for 0–1 knapsack problems", Working paper, Department of Management Science University of Waterloo, 1986.

[8] Sahni, S., "Approximate algorithms for the 0–1 knapsack problem", *Journal of the Association for Computing Machinery* 22 (1975) 115–124.